AMENDMENT TO THE CLAIMS:

This listing of claims will replace all prior versions of claims in the application.

Listing of Claims:

1-6. (Cancelled)

7. (Currently Amended) A method for controlling a function ~~(f)~~ executable by various software products ~~(4a, 4b, 6a, 6b)~~ by means of commands ~~(Pa-Pd)~~ specific to the respective software products and each command capable of having at least one option, the software products being installed in at least one machine ~~(2a, 2b)~~ of a computer system ~~(10)~~, comprising defining in an abstract class an abstract method for the function ~~(f)~~, the abstract method including parameters corresponding to a union, in the logical sense, of all options ~~(Table C)~~ of a specific command, defining a common command ~~(P0)~~ that includes arbitrary symbols corresponding to parameters of the abstract method, creating at least one driver ~~(52)~~ for implementing the abstract method in a machine, and executing by the driver one of the specific commands with options equivalent to the options of the common command.

8. (Currently Amended) A method according to claim 7, wherein equivalence between options of the specific command and options of the common command comprises creating a configuration file ~~(ConfigPrint)~~ defining types and default values of the options of each specific command that can be executed by the driver, and determining parameters of one of said specific commands by consulting a configuration file by means of the common command.

9. (Currently Amended) A method according to claim 7, wherein a driver ~~(52a)~~ corresponds to a machine ~~(2a)~~ of the computer system.

10. (Currently Amended) A method according to claim 8, wherein a driver ~~(52a)~~ corresponds to a machine ~~(2a)~~ of the computer system.

11. (Currently Amended) A method according to claim 7, wherein the abstract class is the most abstract class that can be defined~~, such as an interface in the Java® language~~.

12. (Currently Amended) A method according to claim 8, wherein the abstract class is the most abstract class that can be defined~~, such as an interface in the Java® language~~.

13. (Currently Amended) A method according to claim 9, wherein the abstract class is the most abstract class that can be defined~~, such as an interface in the Java® language~~.

14. (Currently Amended) A method according to claim 10, wherein the abstract class is the most abstract class that can be defined~~, such as an interface in the Java® language~~.

15. (Currently Amended) A method according to claim 7, wherein the abstract class contains at least some of the methods relating to functions of a functionality ~~(F)~~ common to the software products.

16. (Currently Amended) A method according to claim 8, wherein the abstract class contains all or some of the methods relating to functions of a functionality ~~(F)~~ common to the software products.

17. (Currently Amended) A method according to claim 9, wherein the abstract class contains all or some of the methods relating to functions of a functionality ~~(F)~~ common to the software products.

18. (Currently Amended) A method according to claim 10, wherein the abstract class contains all or some of the methods relating to functions of a functionality ~~(F)~~ common to the software products.

19. (Currently Amended) A method according to claim 11, wherein the abstract class contains all or some of the methods relating to functions of a functionality ~~(F)~~ common to the software products.

20. (Currently Amended) A computer system ~~(10)~~ comprising at least one machine having various software products ~~(4a, 4b, 6a, 6b)~~ having in common at least one function ~~(f)~~ executable by means of commands ~~(Pa-Pd)~~ specific to the respective software products and each command capable of having at least one option, and adapted to implement a method for controlling a function ~~(f)~~ executable by various software

products ~~(4a, 4b, 6a, 6b)~~ by means of commands ~~(Pa-Pd)~~ specific to the respective software products and each command capable of having at least one option, the software products being installed in at least one machine ~~(2a, 2b)~~ of a computer system ~~(10)~~, means for defining in an abstract class an abstract method for the function ~~(f)~~, the abstract method including parameters corresponding to a union, in the logical sense, of all options ~~(Table C)~~ of a specific command, means for defining a common command ~~(P0)~~ that includes arbitrary symbols corresponding to parameters of the abstract method, means for creating at least one driver ~~(52)~~ for implementing the abstract method in a machine, and means for executing by the driver one of the specific commands with options equivalent to the options of the common command.

21. (Currently Amended) A computer system, according to claim ~~26~~ 20, further comprising means for creating a configuration file ~~(ConfigPrint)~~ defining the types and the default values of the options of each specific command that can be executed by the driver, and means determining the parameters of one of said specific commands by consulting a configuration file by means of the common command so as to provide equivalence between the options of the specific command and the options of the common command.

22. (Currently Amended) A computer system according to claim 20 wherein a machine ~~(2a)~~ of the computer system includes a drive ~~(52a)~~.

23. (Previously Presented) A computer system according to claim 20 wherein the abstract class is the most abstract class that can be defined.

24. (Currently Amended) A computer system according to claim 20 wherein the abstract class contains all or some of the methods relating to functions of a same functionality ~~(F)~~ common to the software products.

25. (Currently Amended) A computer system as set forth in claim 23 wherein the ~~absract~~ abstract class is an interface in ~~the Java®~~ a programming language.

26. (New) A method for controlling a function executable by various software products by means of commands specific to the respective software products and each command capable of having at least one option, comprising:

defining in an abstract class an abstract method for the function, the abstract method including parameters corresponding to all of the options of a specific command, where the options are an argument that is capable of modifying the function of a specific command;

defining a common command that includes arbitrary symbols corresponding to parameters of the abstract method;

creating at least one driver for implementing the abstract method in a machine; and

executing by the driver one of the specific commands with options equivalent to the options of the common command.

27. (New) The method of claim 7, wherein the options are an argument that is capable of modifying the function of the specific command.

AMENDMENTS TO THE DRAWINGS:

Please substitute the attached replacement sheet for sheet 5/6 of the drawings as originally filed.